Agile Software
Requirements

Matthew Renze
Iowa State University
COMS 409 – Software Requirements
Purpose

• Introduce you to Agile software development
• Discuss Agile software requirements
Overview

• What is Agile?
• Waterfall vs. Agile
• User Stories
• Embedded Documentation
• Non-functional Requirements
• Q & A
About Me

• Independent software consultant
• 14 years of professional Agile software development experience
• Data-driven desktop, server, and web apps
  • Web-based GIS data warehouse
  • Energy data ETL application
  • Global data management system
  • Intelligent lighting control systems
  • Open source data explorer
Education

• BS in Computer Science
• BA in Philosophy
  • Minor in Economics
  • Focus on Artificial Intelligence and Machine Learning
• AS in MIS
• AS in Business Administration
Public Speaking

• Events:
  • Iowa Code Camp
  • Nebraska Code Camp
  • Iowa .NET Users Group
  • Agile Iowa

• Topics:
  • Lean / Agile
  • Data Analysis
  • Patterns, Practices, and Principles
What is Agile?
What is Agile?

• Started with the Agile Manifesto
  • 4 value propositions
  • 12 principles

• Common set of practices across several methodologies

Agile Values

• Individuals and interactions
  • over processes and tools

• Working software
  • over comprehensive documentation

• Customer collaboration
  • over contract negotiation

• Responding to change
  • over following a plan

Source: http://agilemanifesto.org/
12 Principles of Agile

1. Continuous delivery of value
2. Embrace changing requirements
3. Frequent deployment
4. Customer collaboration
5. Motivated individuals
6. Face-to-face conversation
12 Principles of Agile

7. Working software as measure of progress
8. Sustainable development
9. Technical excellence
10. Simplicity
11. Self-organization
12. Continuous improvement
Agile Methodologies

• Scrum
• XP
• Kanban
• Lean
• And many more...

Source: http://parkertoddloesch.files.wordpress.com/2011/09/umbrella.jpg
Waterfall vs. Agile
Waterfall

Start

Requirements

Design

Implementation

Verification

Maintenance

Finish

Time
Agile

# Waterfall vs. Agile Processes

<table>
<thead>
<tr>
<th></th>
<th>Waterfall</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure of Success</td>
<td>Execution of the plan</td>
<td>Working software</td>
</tr>
<tr>
<td>Management Culture</td>
<td>Command and control</td>
<td>Self-organization</td>
</tr>
<tr>
<td>Requirements / Design</td>
<td>Big and upfront</td>
<td>Just-in-time / minimal</td>
</tr>
<tr>
<td>Code / Implementation</td>
<td>Code first and test later</td>
<td>Test and code together</td>
</tr>
<tr>
<td>Testing and QA</td>
<td>Big test plan / test last</td>
<td>Test early / continuously</td>
</tr>
<tr>
<td>Planning and Scheduling</td>
<td>Large detailed plan</td>
<td>Short, iterative increments</td>
</tr>
</tbody>
</table>
User Stories
User Story

• Short description of functionality that will provide value to a user

• Contains:
  • Title
  • Description
  • Acceptance Criteria

• Placeholder for a conversation to occur

<table>
<thead>
<tr>
<th>Enter PIN Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>As an ATM user</td>
</tr>
<tr>
<td>I want to enter my PIN</td>
</tr>
<tr>
<td>So that I can withdraw cash</td>
</tr>
</tbody>
</table>
Title: Enter Personal Identification Number (PIN)

Description:
As an ATM user
I want to enter my PIN
So that I can withdraw cash

Acceptance Criteria:
PIN must be four digits long
PIN must not allow alpha or special characters
PIN must be entered within 30 seconds
or the transaction will be canceled
Agile Requirements

• Feature requests captured in user stories
• User stories are prioritized in product backlog
• Work on user stories in priority order
Agile Requirements

• For each user story:
  • Gather requirements through collaboration
  • Implement functionality using Agile practices
  • Requirements become embedded in code

<table>
<thead>
<tr>
<th>Priority</th>
<th>Product Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Feature 1</td>
</tr>
<tr>
<td></td>
<td>Feature 2</td>
</tr>
<tr>
<td></td>
<td>Feature 3</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>Feature n</td>
</tr>
</tbody>
</table>
Agile Requirements Gathering

- Minimal documentation
- Communication and collaboration are critical
- Uses whole-team approach
- On-site user representation
- Rapid feedback loop is critical to success

Embedded Documentation
Waterfall Requirements Documentation

• In Waterfall:
  • Documentation is the blueprint
  • Code is the product being produced

• Documentation is:
  • Often out of date
  • Not read frequently
  • Not executable

Agile Requirements Documentation

• In Agile:
  • Code is the blueprint
  • Working software is the product being produced

• Code is:
  • Always up to date
  • Continuously read
  • Executable

• Requirements are embedded in the code

## Agile Practices for Embedded Documentation

<table>
<thead>
<tr>
<th>Practice</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test-Driven Development (TDD)</td>
<td>Low-level Behaviors</td>
</tr>
<tr>
<td>Behavior-Driven Development (BDD)</td>
<td>High-level Behaviors</td>
</tr>
<tr>
<td>Domain-Driven Design (DDD)</td>
<td>High-level Policy / Domain Logic</td>
</tr>
<tr>
<td>Domain-Specific Languages (DSL)</td>
<td>Human-readable code for business</td>
</tr>
<tr>
<td>Clean Code</td>
<td>Reader-centered code for devs.</td>
</tr>
</tbody>
</table>
Non-Functional Requirements
Non-Functional Requirements

• Maintainability
• Performance
• Reliability
• Security
• Testability
• Usability
Agile NFRs

Driven by Users:
• Performance
• Security
• Usability

Driven by Developers:
• Maintainability
• Readability
• Testability
Conclusion
Conclusion

• What is Agile?
• Waterfall vs. Agile
• User Stories
• Embedded Documentation
• Non-functional Requirements
Recommended Reading

- Agile Software Requirements
  Lean Requirements Practices for Teams, Programs, and the Enterprise
  Dean Leffingwell
  Foreword by Don Reinertsen

- Software Requirements
  Practical techniques for gathering and managing requirements throughout the product development cycle
  Karl E. Wiegers
Contact Info

Matthew Renze
matthew@renzeconsulting.com

Renze Consulting
www.renzeconsulting.com

Data Explorer
http://www.data-explorer.com
Q & A